PATENT APPLICATION

METHOD AND APPARATUS FOR MASSIVELY PARALLEL CONFIGURATION OF FPGAS

Inventor(s):   Paul Tracy, a citizen of the United States, residing at
735 Bamboo Drive
Sunnyvale, CA 94086

Adam Wright, a citizen of the United States, residing at
435 Milan Court #118
San Jose, CA 95134

Assignee: Altera Corporation
101 Innovation Drive
San Jose, CA, 95134

Entity: Large

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, 8th Floor
San Francisco, California 94111-3834
Tel: 415-576-0200

METHOD AND APPARATUS FOR MASSIVELY PARALLEL CONFIGURATION OF FPGAS

BACKGROUND OF THE INVENTION

5   [0001]   Programmable devices, such as SRAM-based FPGAs, can be rapidly reconfigured to perform many different functions. Typically, programmable devices include a number of different functional units connected by programmable interconnections. The functions of programmable device are determined by configuration data. Configuration data is loaded into the programmable device and defines the configuration of the functional units and the

10   programmable interconnections. This, in turn, defines the overall functions of the programmable device.

[0002]   To test programmable devices, test configuration data is loaded into a programmable device to define one or more test functions. The test functions are then tested to ensure that the programmable device is operating properly. Typically, testing a

15   programmable device requires reconfiguration with thousands of different sets of test configuration data to achieve sufficient test coverage. It is desirable to reduce the time, and consequently the cost, of testing for programmable devices.

[0003]   Loading test configuration data is one of the most time-consuming portions of the test process. Typically, test configuration data must be transferred from a test apparatus into

20   the internal configuration memory of a programmable device. Configuration memory is typically divided into a number of configuration words. Each configuration word defines the configuration of a portion of the programmable device. Typically, a configuration word has far more bits than the number of input pins available for loading configuration data. For example, configuration words may have hundreds or thousands of configuration bits. One

25   prior system for setting the value of each configuration word is to divide each configuration word into a number of configuration blocks. Each configuration block is treated as a large shift register and is assigned to a different input pin. In this system, configuration data is serially loaded into each configuration block. This system can take thousands of clock cycles to load configuration data for a single test, making testing time-consuming and expensive.

30   [0004]   Another prior system for loading test configuration data divides the configuration word into a number of configuration blocks. This system loads configuration data into a

configuration block in parallel. Each configuration block is simultaneously loaded with identical configuration data. If each configuration block has the same number of bits as there are input pins, then configuration data for an entire configuration word can be loaded in a single clock cycle.

5   [0005] This system's "all or nothing" approach to repeatability leads to problems. If different configuration data needs to be loaded into different blocks, this system cannot be used. This can often occur when configuration data is not symmetrical on a block-wise basis, or when the programmable device architecture is asymmetrical. In these situations, another system for loading configuration data, such as the serial loading system discussed above must

10   be used. Thus, when configuration data violates block-wise symmetry in even a minor way, the system punishes this digression with a maximum load cost. This is especially inefficient because typical configuration data includes a large number of 0's interspersed between a few 1's.

[0006] It is desirable to improve the efficiency of loading configuration data into

15   programmable devices with asymmetrical configuration data. It is further desirable to allow for loading of configuration data using one of several alternate systems to maximize the efficiency of the loading process.

BRIEF SUMMARY OF THE INVENTION

20   [0007] The invention, generally, is a method and system for compressing configuration data to decrease the time spent loading configuration data into a programmable device. In an embodiment, the programmable device includes a configuration word register comprising a plurality of configuration blocks, a plurality of configuration inputs selectively coupled with each of the plurality of configuration blocks, and a plurality of command inputs. The

25   command inputs are adapted to independently enable one or more of configuration blocks to simultaneously load configuration data via the plurality of configuration inputs.

[0008] In one embodiment, each of the configuration blocks is coupled with one of the plurality of command inputs. Further, at least one configuration block has the same number of bits as there are configuration inputs. In an alternate embodiment, one or more

30   configuration blocks have less bits than the number of configuration inputs.

2

[0009] In another embodiment, the programmable device has a configuration memory with a plurality of memory locations. The configuration memory is coupled with the configuration word register and is adapted to load configuration data from the configuration word register.

[0010] In yet a further embodiment, the programmable device includes a configuration mode input and a configuration controller coupled with the configuration mode input. The configuration controller enables the configuration blocks to simultaneously load configuration data via the configuration inputs in response to an enabling by the command inputs in response to a first state of the configuration mode input. In response to a second state of the configuration mode input, the configuration controller enables the loading of configuration data into the configuration word register via an alternate coupling with configuration data.

[0011] The alternate coupling with configuration data is via the plurality of configuration inputs, or in another embodiment, via the plurality of command inputs. In a further embodiment, the alternate coupling with configuration data simultaneously loads one bit of configuration data into each of the configuration blocks.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The invention will be described with reference to the drawings, in which:

Figure 1 illustrates a prior system for loading configuration data into a programmable device;

Figure 2 illustrates another prior system for loading configuration data into a programmable device;

Figure 3 illustrates a system for loading configuration data into a programmable device according to an embodiment of the invention; and

Figure 4 illustrates a system for loading configuration data into a programmable device according to another embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0013] Figure 1 illustrates a prior system for loading configuration data into a programmable device 100. Programmable device 100 includes a configuration word register 105 adapted to receive configuration data via test pin inputs 120. Configuration word register

3

105 is comprised of N number of configuration blocks, such as configuration blocks 110, 111, 112, 113, 114, and 115. In programmable device 100, each configuration block is connected with one of the test pin inputs 120. Each configuration block is also connected with clock 125.

5 **[0014]** Each configuration block has M number of configuration bits. As M is typically greater than 1, programmable device 110 must load configuration data into each configuration block serially via the test pin inputs 120. The highest order, or M, bit of configuration data for each configuration block is placed on the corresponding test pin inputs 120. In response to a signal from clock 125, the M bits of configuration data for each

10 configuration block are loaded into their respective configuration blocks from the test pin inputs 120.

[0015] Next, the second-to-highest order, or M-1, bits for each configuration block are similarly placed on their corresponding test pin inputs 120. In response to clock 125, the M-1 bits of configuration data for each configuration block are loaded from the test pin inputs 120

15 into the appropriate configuration blocks. As this occurs, the M bits previously loaded are shifted one position to the left in the configuration word register 105. This process is repeated for all M bits of each configuration block.

[0016] Once all the configuration blocks in configuration word register 105 are loaded with the configuration data, the configuration data is transferred by output connections 130 to

20 configuration memory 135. There may be up to M x N total output connections 130 between the configuration memory 135 and the configuration word register 105. Configuration memory 135 is a memory device, such as a SRAM, for storing the complete set of configuration of the programmable device. Typically, configuration memory includes a number of different configuration words. The configuration data received from the

25 configuration word register 105 is stored in the appropriate location in configuration memory 135. As an example, a programmable device may require hundreds or thousands of configuration words to specify a given configuration.

[0017] In programmable device 100, serially loading configuration data into each configuration block and then transferring this data to the configuration memory takes at least

30 M clock cycles. For example, with a 1000 bit configuration word and 50 test pin inputs (i.e. N=50 and M = 1000/N = 20), each configuration word takes at least 20 clock cycles to load.

[0018] Figure 2 illustrates another prior system for loading configuration data into

4

programmable device 200 that consumes less time. Programmable device 200 includes a configuration word register 205. Configuration word register 205 is divided into N number of configuration blocks of up to M bits each. There are also M number of test pin inputs 210. Each bit of a configuration block is connected in parallel with the corresponding bits in the other configuration blocks and to one of the M test pin inputs 210.

[0019] Programmable device 200 loads configuration data in parallel via the test pin inputs 210. An M number of bits of configuration data are placed on the test pin inputs 210. In response to a signal from clock 220, the M bits of configuration data are loaded into each of the configuration blocks of configuration word register 205. The configuration data loaded into the configuration word register 205 is then transferred to a configuration memory (not shown) via output connections 215, similar to that discussed above.

[0020] Programmable device 200 is capable of loading an entire configuration word in a single clock cycle. However, programmable device must load the same configuration data into each of its configuration blocks. Thus, although programmable device 200 is very efficient in loading configuration data with block-wise symmetry, it cannot load all types of configuration data.

[0021] Figure 3 illustrates a system for loading configuration data into a programmable device 300 according to an embodiment of the invention. Programmable device 300 includes a configuration word register 305. Configuration word register includes an N number of configuration blocks, for example, configuration blocks 330, 331, 332, 333, 334, and 335. N can be any positive whole number and the number of configuration blocks in Figure 3 should be viewed as an illustrative example and not a limitation.

[0022] In an embodiment, each configuration block includes M configuration bits. In an alternate embodiment, configuration blocks can have a different number of configuration bits ranging from 1 to M. This alternate embodiment can be used in programmable devices where the total number of bits, T, in configuration word register 305 is not evenly divisible by M. In an example, there can be N-1 number of configuration blocks with M bits each and one configuration block with T modulo M bits. Other ways of allocating configuration bits between configuration blocks are also possible in this alternate embodiment.

[0023] Each bit of each configuration block is connected with one of the set of M number of test pin inputs 310, such that M bits of configuration data can be loaded into each configuration block in parallel. If a configuration block has less than M bits, then some of

the test pin inputs will not be connected with that configuration block; however, each configuration block will be connected with at least one test pin input.

[0024]   The configuration blocks are connected with clock 320. Each configuration block in configuration word register 305 is also separately connected with one of a set of block enable lines 315. If there are N configuration blocks, there will be N block enable lines in the set of block enable lines 315. When a block enable line is asserted while a clock signal is received, the corresponding configuration block will load up to M bits of configuration data from the set of test pin inputs 310.

[0025]   In an embodiment, programmable device 300 receives configuration data in the form of pairs of command words and data words. Each data word includes up to M bits of configuration data. Each command word indicates one or more configuration blocks that will load the configuration data from the data word. In a further embodiment, each command word has N bits. Each bit of the command word corresponds to one of the block enable lines associated with one of the configuration blocks. If a bit of a command word is a "1", the corresponding block enable line is to be asserted and the configuration block will load up to M bits of configuration data via test pin inputs 310. Conversely, if a bit of a command word is a "0", the corresponding block enable line will not be asserted and the configuration block will ignore the configuration data from test pin inputs 310. If a configuration block has less than M bits, then the configuration block will ignore the excess number of bits from test pin inputs 310.

[0026]   An example loading operation of programmable device 300 begins by placing the bits of the command word on the corresponding lines of the set of block enable lines 315. The bits of the data word are placed on the test pin inputs 310. Upon receiving a signal from clock 320, up to M bits of configuration data are simultaneously loaded into the configuration blocks having asserted block enable lines. The number of configuration blocks simultaneously loaded with configuration data can be any number from one up to N. The loading operation can be repeated numerous times with different command and data words until all of the configuration data for a given configuration word has been loaded into the configuration word register 305. In an embodiment, the complete configuration word is then transferred to a configuration memory (not shown) via output connections 325. This loading operation can then be started again for the next configuration word, until the configuration of the programmable device 300 is complete.

[0027] If the configuration blocks in configuration word register 305 is aligned with the symmetry in command words, the programmable device 300 will be able to load many configuration blocks with identical configuration data. This results in a substantial decrease in configuration time. Moreover, programmable device 300 can maximize the use of symmetry in a configuration word by simultaneously loading as many configuration blocks as possible with successive blocks of configuration data until the configuration word register is completely filled. Additionally, programmable device 300 can load asymmetrical configuration data by only asserting one block enable line at a time.

[0028] In some cases, if the configuration data in a configuration word is very asymmetrical, it may be more efficient to load configuration data serially, as discussed above with reference to Figure 1. For example, if N, the number of configuration blocks, is greater than M, the maximum number of bits in a single configuration block, and there are more than M different blocks of configuration data in a configuration word, then it is more efficient to load the configuration word serially. In a further embodiment of the invention, a programmable device can alternate between the loading system discussed with reference to Figure 3 and an alternate loading system, such as that discussed with reference to Figure 1.

[0029] Figure 4 illustrates a system for loading configuration data into programmable device 400 according to this further embodiment of the invention. Programmable device 400 includes a configuration word register 410. Configuration word register includes an N number of configuration blocks, for example, configuration blocks 411, 412, 413, and 414. N can be any positive whole number and the number of configuration blocks in Figure 4 should be viewed as an illustrative example and not a limitation.

[0030] In an embodiment, each configuration block includes M configuration bits. Similar to that discussed above, an alternate embodiment of configuration word register 410 can have configuration blocks with differing numbers of configuration bits ranging from 1 to M. Each bit of each configuration block is connected with one of the set of M number of test pin inputs 455, such that M bits of configuration data can be loaded into each configuration block in parallel, similar to that described above.

[0031] Each configuration block in configuration word register 410 is also separately connected with one of a set of block enable lines, such as block enable lines 460, 465, 470, and 475. If there are N configuration blocks, there will be N block enable lines in the set of block enable lines. When a block enable line is asserted while a clock signal is received via a

clock (not shown), the corresponding configuration block will load up to M bits of configuration data from the set of test pin inputs 455.

[0032] Each configuration block is also connected with one of a set of serial test pin inputs, such as serial test pin inputs 435, 440, 445, and 450. Each serial test pin input loads one bit at a time into its associated configuration block. As discussed above, as each new configuration bit is loaded into a configuration block, the previously loaded bits are shifted to the next bit position (for example, bits can be shifted left or right).

[0033] A configuration controller 405 determines whether configuration data should be loaded serially or in parallel. Configuration controller 405 is connected with mode pin 420, a set of M data inputs 425, and a set of N command inputs 430. In an embodiment, when mode pin is asserted, configuration controller 405 connects data inputs 425 with test pin inputs 455 and each of the set command inputs 430 with a corresponding one of the block enable lines. In this manner, each configuration block with an asserted enable line will load configuration data from test pin inputs 455. This process is repeated for each different block of configuration data in a configuration word.

[0034] Conversely, when mode pin 420 is not asserted, configuration controller 405 connects each of the set of command inputs 425 with a corresponding one of the serial test pin inputs. In this manner, each configuration block can simultaneously load 1 bit of configuration data. This process is repeated for a total M times to load a complete configuration word. In this embodiment, the N command inputs 430 are used to load configuration data for serial data transfer, while the M data inputs 425 are used to load configuration data in parallel. This is done because there is N number of serial test pin inputs.

[0035] It should be noted that a programmable device can select either of the two loading systems for each configuration word in a programmable device's configuration to maximize the efficiency of the configuration process. In an embodiment, a test apparatus or other configuration device evaluates each configuration word before it is to be loaded into a configuration device to determine the optimal loading system. In a further embodiment, the parallel loading system is used when a configuration word has more than M number of unique block patterns of configuration data. Otherwise, the serially loading system, which takes M number of cycles to load a configuration word regardless of its content, is used.

[0036]   In another embodiment of system 400, both the N command inputs 430 and the M data inputs 425 are used to together to load configuration data for serial data transfer. In this embodiment, the configuration blocks are different sizes depending on whether the parallel data transfer system or the serial data transfer system is used. During parallel data transfer, there are N configuration blocks of M bits each. For serial data transfer, there are M+N configuration blocks of $T/(M+N)$ bits each, where T is the total number of bits in configuration word register 410. There are also M+N serial test pin inputs associated with the M+N configuration blocks.

[0037]   The configuration controller 405 connects the N command inputs 430 with the block enable lines during parallel loading, as described above, and with N of the M+N serial test pin inputs during serial loading. Similarly, the configuration controller 405 connects the M data inputs 425 with test pin inputs 455 during parallel loading, as described above, and with M of the M+N serial test pin inputs during serial loading. In this embodiment, serial loading of a configuration word takes $T/(M+N)$ clock cycles. When there are less than $T/(M+N)$ different M-sized blocks of configuration data in a configuration word, the parallel loading system is used; otherwise, the serial loading system is more efficient.

[0038]   This invention allows for greatly reduced configuration times, which is particularly advantageous when testing programmable devices by loading a large number of different configurations. Because of the decreased time, and consequently cost, of testing, it allows for more asymmetrical designs, which often are better for design routing purposes, to be used. Although the invention has been discussed with respect to specific examples and embodiments thereof, these are merely illustrative, and not restrictive, of the invention. For instance, the embodiment of Figure 4 is adaptable to other loading systems besides that described with reference to Figure 1. Thus, the scope of the invention is to be determined solely by the claims.